

**Algoritmusok tervezése és elemzése**

**Stumpf Máté Benedek, HWS19X**

**Spam filter**

A problémám a manapság egyre gyakoribb spam emailek és online csalások megelőzése. E célból készítettem egy spam szűrő programot, amely bemenetként fogad egy akár több soros email szöveget. Ez a szöveg bemásolható a terminálba (***másolás után nyomjuk meg a Ctrl + Z, vagy a Ctrl + D billentyűkombinációt***), ahol a program elemzi, hogy tartalmaz-e előre meghatározott spam-re utaló kulcsszavakat vagy URL-eket.

A program feldolgozza az email tartalmát, a sorvégeket szóközökre cseréli, majd kereső algoritmusokkal azonosítja a kulcsszavakat és az URL-eket. Az eredményeket megjeleníti: felsorolja a kulcsszavakat azok előfordulási számával együtt, valamint a megtalált URL-eket, és ellenőrzi, hogy azok tanúsítványa biztonságos-e (a „http://” használata alapján).

A program végül eldönti, hogy az email spam-e, és erről tájékoztatja a felhasználót. A kulcsszókereséshez, az URL-ek felismeréséhez és az eredmények kiértékeléséhez kereső, eldöntő és megszámláló algoritmusokat alkalmaztam.

A következő oldalon található a pszeudokód, és azt követően majd a program forráskódja, Python nyelven. (A tartalomjegyzék hyperlinkeket tartalmaz, hogy megkönnyítse a fejezetek közti keresést.)

Tartalomjegyzék:

[***Pszeudokód:*** 2](#_Toc185540297)

[***A program forráskódja:*** 3](#_Toc185540298)

[***Itt található pár példa a kimenetre:*** 4](#_Toc185540299)

[***Végszó:*** 5](#_Toc185540300)

***Pszeudokód:***

DEFINIÁLD SPAM\_KULCSSZAVAK-AT, MINT SZTRINGEK LISTÁJA  
DEFINIÁLD URL\_MINTA-T, MINT REGEX  
  
FÜGGVÉNY BEMENET\_OLVASÁSA()  
 ÍRD KI: "Kérlek, másold be a szöveget (több soros is lehet). Ha végeztél, nyomd meg a Ctrl+D-t vagy a Ctrl+Z-t:"  
 OLVASD BE az összes sort a standard bemenetről  
 TÁVOLÍTSD EL az új sorokat, helyettesítsd őket szóközökkel  
 ADJ VISSZA a megtisztított szöveget  
  
FÜGGVÉNY SPAM\_FILTER(szöveg)  
 alakítsd át a szöveget kisbetűssé  
 inicializáld talalt\_kulcsszavak-at, mint üres szótár  
 inicializáld talalt\_url-eket, mint üres lista  
  
 MINDEN egyes kulcsszóra a spam\_kulcsszavak-ban  
 SZÁMOLD MEG a kulcsszó előfordulásait a szövegben  
 HA az előfordulások száma > 0  
 ADD HOZZÁ a kulcsszót és az előfordulások számát a talalt\_kulcsszavak-hoz  
  
 KERESS MEG minden egyezést az url\_minta alapján a szövegben  
 ADD HOZZÁ az egyezéseket a talalt\_url-ekhez  
  
 inicializáld nem\_biztonsagos\_url-t, mint üres lista  
 MINDEN URL-re a talalt\_url-ekben  
 HA az URL "http://"-el kezdődik  
 ADD HOZZÁ az URL-t a nem\_biztonsagos\_url-hez  
  
 ELLENŐRIZD, hogy a talalt\_kulcsszavak vagy a talalt\_url nem üresek-e  
 ADJ VISSZA (spam\_e, talalt\_kulcsszavak, talalt\_url, nem\_biztonsagos\_url)  
  
FÜGGVÉNY MAIN()  
 HÍVD MEG a bemenet\_olvasása() függvényt, és tárold az eredményt a szöveg változóban  
  
 HÍVD MEG a spam\_filter(szöveg) függvényt, és tárold az eredményeket (spam\_e, talalt\_kulcsszavak, talalt\_url, nem\_biztonsagos\_url)-ként  
  
 HA spam\_e  
 ÍRD KI: "Az email valószínűleg SPAM!"  
 KÜLÖNBEN  
 ÍRD KI: "Az email valószínűleg NEM SPAM."  
  
 HA talalt\_kulcsszavak nem üresek  
 ÍRD KI: "Talált spam-re utaló kulcsszavak és előfordulásaik:"  
 MINDEN kulcsszó és szám párra a talalt\_kulcsszavak-ban  
 ÍRD KI a kulcsszót és az előfordulások számát  
  
 HA talalt\_url nem üresek  
 ÍRD KI: "Talált gyanús URL-ek:"  
 MINDEN URL-re a talalt\_url-ekben  
 ÍRD KI az URL-t  
  
 HA nem\_biztonsagos\_url nem üresek  
 ÍRD KI: "Figyelmeztetés: A következő URL-ek tanúsítványa nem biztonságos:"  
 MINDEN URL-re a nem\_biztonsagos\_url-ben  
 ÍRD KI az URL-t  
  
HÍVD MEG MAIN()  
  
VÉGE

# ***A program forráskódja:***

import re

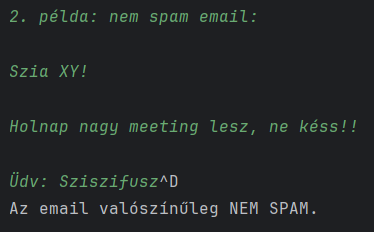
import sys  
  
# Gyakori spam kulcsszavak és kifejezések  
spam\_kulcsszavak = ["nyeremény","nyereményjáték", "ingyen", "kattints ide", "ajánlat", "gyors pénz", "növeld a bevételed", "pénzkeresési lehetőség"]  
  
# URL-ek keresésére használt minta  
url\_minta = r"http[s]?://(?:[a-zA-Z]|[0-9]|[$-\_@.&+]|[!\*\\(\\),]|(?:%[0-9a-fA-F][0-9a-fA-F]))+|(?:[a-zA-Z0-9-]+\.)+[a-zA-Z]{2,}"  
  
def bemenet\_olvasasa():  
 #Beolvassa a teljes bemenetet és megtisztítja.  
 print("Kérlek, másold be a szöveget (több soros is lehet). Ha végeztél, nyomd meg a Ctrl+D-t (pycharm IDE) vagy a Ctrl+Z-t:")  
 bemenet = sys.stdin.read() # Beolvassa az összes sort nyers bemenettel, így tudunk több sort egyszerre bevinni  
 return bemenet.strip().replace("\n", " ") # A sortöréseket szóközzé tesszük  
  
def spam\_filter(szoveg):  
  
 #Ellenőrzi, hogy a szöveg spam-e, és visszaadja a talált gyanús elemeket (kulcsszavak és URL-ek).  
  
 szoveg = szoveg.lower() # Kisbetűsre alakítjuk a szöveget  
 talalt\_kulcsszavak = {} #kulcsszavak tárolásához szükséges lista  
  
  
 # Kulcsszavak keresése és a megjelenésük megszámlálása  
 for kulcsszo in spam\_kulcsszavak:  
 elofordulas = szoveg.count(kulcsszo)  
 if elofordulas > 0:  
 talalt\_kulcsszavak[kulcsszo] = elofordulas  
  
 # URL-ek keresése  
 talalt\_url = re.findall(url\_minta, szoveg)  
  
 # Ellenőrizzük, hogy van-e 'http' URL és figyelmeztessünk a nem biztonságos tanúsítványra  
 nem\_biztonsagos\_url = [url for url in talalt\_url if url.startswith("http://")]  
  
 # A szöveg spam, ha találunk kulcsszavakat vagy URL-eket  
 is\_spam = bool(talalt\_kulcsszavak or nem\_biztonsagos\_url)  
  
 return is\_spam, talalt\_kulcsszavak, talalt\_url, nem\_biztonsagos\_url  
  
def main():  
 # Bemenet beolvasása és tisztítása  
 szoveg = bemenet\_olvasasa()  
  
 # Spam szűrés  
 is\_spam, talalt\_kulcsszavak, talalt\_url, nem\_biztonsagos\_url = spam\_filter(szoveg)  
  
 # Eredmény kiírása  
 if is\_spam:  
 print("Az email valószínűleg SPAM!")  
 else:  
 print("Az email valószínűleg NEM SPAM.")  
  
 # Talált kulcsszavak és URL-ek listázása  
 if talalt\_kulcsszavak:  
 print("Talált spam-re utaló kulcsszavak és előfordulásaik:")  
 for kulcsszo, db in talalt\_kulcsszavak.items():  
 print(f"- {kulcsszo}: {db} alkalommal")  
  
 #itt az URL-eket akkor listázza ki ha az email spamnek minősül  
 if talalt\_url and is\_spam:  
 print("Talált gyanús URL-ek:")  
 for url in talalt\_url:  
 print(f"- {url}")  
  
 # Nem biztonságos http URL-ek figyelmeztetése  
 if nem\_biztonsagos\_url:  
 print("Figyelmeztetés: A következő URL-ek tanúsítványa nem biztonságos:")  
 for url in nem\_biztonsagos\_url:  
 print(f"- {url}")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

# ***Itt található pár példa a kimenetre:***

A screen shot of a computer

Description automatically generated

***Nem spam email-re adott kimenet:***

******

# ***Végszó:***

Azért ezt a problémát választottam, mivel engem főképpen a kiberbiztonság érdekel, és a későbbiekben ebben a szférában szeretnék elhelyezkedni.  
  
A programot a PyCharm nevű IDE-ben fejlesztettem.

***Github link a projekthez:***

<https://github.com/Y0r0zuya/Spam_Filter>

***Készítette: Stumpf Máté Benedek – HWS19X  
Tantárgy: Algoritmusok tervezése és elemzése  
Kurzusfelelős: Dr. Túri József***